Deinterleaving Markov Processes: the Finite-Memory Switch Case

Gadiel Seroussi
Hewlett-Packard Laboratories
Palo Alto, CA, USA
Email: gseroussi@ieee.org

Wojciech Szpankowski
Purdue University,
West Lafayette, IN, USA
Email: spa@cs.purdue.edu

Marcelo J. Weinberger
Hewlett-Packard Laboratories
Palo Alto, CA, USA
Email: marcelo.weinberger@hp.com

***Abstract*—We study the problem of deinterleaving a set of finite-memory (Markov) processes over disjoint finite alphabets, which have been randomly interleaved by a finite-memory switch, extending previous results obtained for the case of a memoryless switch [1]. The deinterleaver has access to a sample of the resulting interleaved process, but no knowledge of the number or structure of the Markov processes, or of the switch. We study conditions for uniqueness of the interleaved representation of a process, showing that certain switch configurations can cause ambiguities in the representation, in addition to those caused by memoryless component processes, which were known in the memoryless switch case. We show that a deinterleaving scheme based on minimizing a penalized maximum-likelihood cost function is strongly consistent also in the finite-memory switch case, in the sense of reconstructing, almost surely as the observed sequence length tends to infinity, a set of component and switch Markov processes compatible with the original interleaved process. Furthermore, under certain conditions on the structure of the switch, we show that the scheme recovers *all* possible interleaved representations of the original process. Experimental results are presented demonstrating that the proposed scheme performs well in practice, even for relatively short input samples.**

I. Introduction

Problems in applications such as data mining, computer security, and genomics, often require the identification of streams of data from different sources, which may be intermingled or hidden (sometimes purposely) among other unrelated streams. The source identification problem studied in this paper is motivated by these applications (more detailed descriptions of the applications can be found in [2], [3]).

Let $A_1, A_2, \ldots, A_m$ be finite, nonempty, disjoint alphabets, let $\mathcal{A} = A_1 \cup A_2 \cup \cdots A_m$, and $\Pi = \{A_1, A_2, \ldots, A_m\}$. We refer to the A_i as *subalphabets*, and to Π as a *partition*, of $\mathcal{A}$. For $a \in \mathcal{A}$, we denote by $\mathbf{A}_\Pi(a)$ the alphabet $A_i \in \Pi$ such that $a \in A_i$. The notation is extended to strings; for a string u^t over $\mathcal{A}$, we write $\mathbf{A}_\Pi(u^t) = \mathbf{A}_\Pi(u_1), \mathbf{A}_\Pi(u_2), \ldots, \mathbf{A}_\Pi(u_t)$. Also, for $\mathcal{A}' \subseteq \mathcal{A}$, and a string u over $\mathcal{A}$, we denote by $u[\mathcal{A}']$ the string over $\mathcal{A}'$ obtained by deleting from u all symbols that are not in $\mathcal{A}'$. Consider m independent, finite-memory (Markov) *component processes* $P_1, P_2, \ldots, P_m$, defined, respectively, over $A_1, A_2, \ldots, A_m$, and of respective orders $k_1, k_2, \ldots, k_m$, i.e., for $1 \le i \le m$ and any sequence $x^t = x_1 x_2 \ldots x_t \in (A_i)^t$, $t > k_i$, we have $P_i(x_t|x^{t-1}) = P_i(x_t|x_{t-k_i}^{t-1})$, and we assume that k_i is the least integer for which this property holds. Consider also a finite-memory *switch process* P_{w}, of order k_{w}, independent of the component processes, and defined over the alphabet Π.[1] We define the *interleaved Markov process* (IMP) $P \triangleq \mathcal{I}_\Pi(P_1, P_2, \ldots, P_m; P_{\mathrm{w}})$ as follows: Given $z^t \in \mathcal{A}^t$, $t \ge 1$, and assuming $\mathbf{A}_\Pi(z_t)=A_i$, we have

$$P(z_t|z^{t-1}) = P_{\mathrm{w}}(A_i|\mathbf{A}_\Pi(z^{t-1}))P_i(z_t|z^{t-1}[A_i])\,. \quad (1)$$

For simplicity, in (1), we assume that every conditioning string over A_i (including the empty string λ), defines a state of P_i, and similarly for P_{w}. This is accomplished, for instance, by defining an arbitrary but fixed initial state for each process.[2] Together with the convention $P(\lambda) = 1$, these assumptions and (1) completely define the process P. Intuitively, at each time instant t, a subalphabet $A_i \in \Pi$ is selected according to P_{w}, which updates its state, and the next output sample z_t is selected from A_i according to the corresponding process P_i, which also updates its state. The states of the other component processes remain unchanged.

Given a sample z^n from P, the problem of interest is to reconstruct the alphabet partition Π, and, consequently, the original sequences from the Markov processes, and the sequence of switch selections. As a byproduct of our interleaving scheme, we will also recover the orders of all the finite-memory processes involved.

We refer to $\mathcal{I}_\Pi(P_1, P_2, \ldots, P_m; P_{\mathrm{w}})$ also as an *IMP representation* of P. We say that a partition Π' of $\mathcal{A}$ is *compatible* with P, denoted $\Pi' \sim P$, if there exist finite-memory processes $P'_1, P'_2, \ldots, P'_{m'}, P'_{\mathrm{w}}$ such that $P = \mathcal{I}_{\Pi'}(P'_1, P'_2, \ldots, P'_{m'}; P'_{\mathrm{w}})$. An IMP may be compatible with more than one partition, i.e., we may have $\Pi \sim P$ and $\Pi' \sim P$ with $\Pi \ne \Pi'$. We refer to this situation as an *ambiguity* in the IMP representation of P. For conciseness, in the sequel, unless specified otherwise, $P = \mathcal{I}_\Pi$ will denote the IMP $\mathcal{I}_\Pi(P_1, P_2, \ldots, P_m; P_{\mathrm{w}})$, with $\Pi = \{A_1, A_2, \ldots, A_m\}$, and

[1] Markov processes are assumed to be time-homogeneous and ergodic, but not necessarily stationary, as we assume fixed initial states. Marginal probability notations (e.g., $P_i(u)$ for a string u) represent the steady-state probabilities of their arguments. We further assume that all symbols $a \in \mathcal{A}$ (and thus all subalphabets $A \in \Pi$) occur infinitely often, and their steady state marginal probabilities are positive.

[2] The simplifying assumption on the initial states is not essential, and all the results of the paper carry to the cases of more general initial state distributions. Given our ergodicity assumptions, the initial conditions of the system are, in effect, immaterial to the main results of the paper.

$P' = \mathcal{I}_{\Pi'}$ will denote the IMP $\mathcal{I}_{\Pi'}(P'_1, P'_2, \ldots, P'_{m'}; P'_{\rm w})$, with $\Pi = \{A'_1, A'_2, \ldots, A'_{m'}\}$.

The deinterleaving problem was studied, for the case of memoryless switches, in [1], where a complete characterization of ambiguities for this case was presented, and a deinterleaving scheme based on minimizing a penalized maximum-likelihood (ML) cost function was proposed, and shown to be strongly consistent. In this paper, we extend the results of [1] to general finite-memory switches. We first study, in Section III, the issue of representation uniqueness and ambiguities. We show that the ambiguity landscape is significantly more intricate in the finite-memory switch case, as ambiguities may arise from a so-called *alphabet domination* phenomenon that cannot occur in the memoryless switch case, in addition to ambiguities due to memoryless component processes, which were identified in [1]. Even when focusing on the latter type of ambiguity, the characterization turns out to be more complex for switches with memory, as ambiguous representations involving memoryless processes can arise from parameter dependencies in the switch. In Section IV we present our deinterleaving scheme, which, as in [1], is based on minimizing a penalized ML cost function. We show that the scheme is strongly consistent in the sense of producing, almost surely as $n\to\infty$, a partition compatible with P. In the *domination-free* case (i.e., when the switch is such that alphabet domination does not arise), the scheme produces, almost surely, a *canonical* partition from which all IMP representations of P can be derived. In Section V, we present experimental results showing that the proposed scheme performs well in practice, even for relatively short input samples.

The deinterleaving problem for the case where all processes involved are of order one had been previously studied in [2], where an approach was proposed that could identify a valid IMP presentation of P with high probability as $n\to\infty$ (the approach as described cannot identify multiple valid solutions when they exist). For the case of a switch with memory, the scheme of [2] imposes the condition that every state of the switch have a self transition of positive probability. This condition implicitly eliminates alphabet domination, but is stronger than needed to do so, or to preclude ambiguities. An experimental comparison of ML-based deinterleaving with the scheme of [2] for the memoryless switch case was presented in [1], showing a much faster convergence for the ML-based scheme. As [2] does not specify the choices of certain crucial practical parameters, an implementation of the scheme for switches with memory was not readily available, and a similar comparison is omitted here.

II. Additional definitions, notation

We denote by $\mathrm{ord}(P_M)$ the minimal order of a finite-memory process P_M over an alphabet A, and by $\mathcal{S}(P_M)$ the set of states of P_M, represented as k-tuples over A, where $k = \mathrm{ord}(P_M)$. We allow for some conditional probabilities $P_M(u_k|u_0^{k-1})$ to be zero. Thus, some k-tuples u^k may be non-reachable (i.e., $P_M(u^k) = 0$); such k-tuples are excluded from $\mathcal{S}(P_M)$ (all states are assumed to be reachable and recurrent).

Consider the IMP $P = \mathcal{I}_\Pi(P_1, P_2, \ldots, P_m; P_{\rm w})$. We define the vector $\mathbf{k} = (k_1, k_2, \ldots, k_m, k_{\rm w})$, where $k_j = \mathrm{ord}(P_j)$ for $j \in \{1, 2, \ldots, m, {\rm w}\}$; we refer to $\mathbf{k}$ as the *order vector* of P.

We will generally denote sequences (or strings) over $\mathcal{A}$ with lower case letters, e.g., $u \in \mathcal{A}^*$, and sequences over Π with upper case letters, e.g., $U \in \Pi^*$. For a sequence $U^n \in \Pi^n$, and a sequence $u^n \in \mathcal{A}^n$, we say that u^n is *consistent* with U^n if $u_i \in U_i$, $1 \le i \le n$. Clearly, if $P_{\rm w}(U^n) > 0$, there exist sequences u^n consistent with U^n with $P(u^n) > 0$; conversely, every sequence u^n with $P(u^n) > 0$ defines a sequence $U^n = \mathbf{A}_\Pi(u^n) \in \Pi^n$, such that u^n is consistent with U^n and $P_{\rm w}(U^n) > 0$.

III. Uniqueness of IMP representations

A. Alphabet domination

Let A, B be arbitrary subalphabets in Π. We say that A *dominates* B (relative to $P_{\rm w}$) if there exists a positive integer L such that for all $U \in \Pi^*$, if $P_{\rm w}(U) > 0$, then $U[\{A, B\}]$ does not contain any run of more than L consecutive occurrences of B. In other words, $P_{\rm w}$ is such that if we have seen L occurrences of B without seeing one of A, then *with probability one* $P_{\rm w}$ will emit an occurrence of A before it emits another occurrence of B. We denote the domination relation of A over B as $A \sqsupset B$, dependence on $P_{\rm w}$ being understood from the context; when A does not dominate B, we write $A \not\sqsupset B$ (thus, for example, $A \not\sqsupset A$). We say that A is *dominant* (in Π, relative to $P_{\rm w}$) if either $m = 1$ (i.e., $\Pi = \{A\}$) or $A \sqsupset B$ for some $B \in \Pi$, and that A is *totally dominant* if either $m = 1$ or $A \sqsupset B$ for *all* $B \in \Pi \setminus \{A\}$. If $A \sqsupset B$ and $B \sqsupset A$, we say that A and B are in *mutual domination*, and write $A \sqsupset\!\sqsubset B$. It is readily verified that domination is an irreflexive transitive relation. When no two alphabets are in mutual domination, the relation defines a strict partial order (see, e.g., [4]) on the finite set Π. We shall make use of the properties of this strict partial order in the sequel.

Domination can occur only if some transition probabilities in $P_{\rm w}$ are zero (therefore, it never occurs when $P_{\rm w}$ is memoryless). It is readily verified, for instance, that if $\mathrm{ord}(P_{\rm w})$=1 and $P_{\rm w}(A|A)$>0 for all A∈Π, then no domination arises. However, this condition, which was imposed in [2], is too stringent to eliminate domination, and as a condition for uniqueness. More examples of domination and its effect on ambiguities will be presented in Example 1 below.

The main properties of the domination relation are formally studied in the full version of the paper. The following informal statement about an IMP $P = \mathcal{I}_\Pi$ summarizes some important properties, which derive immediately from our ergodicity and independence assumptions, and are drawn upon repeatedly in our study of IMP ambiguities.

Fact 1: If $A_1 \not\sqsupset A_2$, the interleaved system can always take a trajectory (of positive probability) where it reaches an arbitrary state s of P_1, and then, without returning to A_1, visits any desired part of A_2 any desired number of times (while the state of P_1 remains, of course, unchanged). The last part of the trajectory is independent of s, or even of the fact that A_1 was visited in the first part.

B. Conditions for uniqueness

In this subsection, we derive sufficient conditions for the uniqueness of IMP representations, and show how ambiguities may arise when the conditions are not satisfied.

The following terminology will help in the discussion of non-unique IMP representations. Let Π and Π' be partitions of $\mathcal{A}$. We say that two alphabets $A_i, A_j \in \Pi$ *share* an alphabet $A'_\ell \in \Pi'$ if A'_ℓ intersects both A_i and A_j. We say that $A_i \in \Pi$ *splits* in Π' if $A'_j \subseteq A_i$ whenever $A'_j \in \Pi'$ and $A'_j \cap A_i \neq \phi$ (thus, A_i is partitioned into subalphabets in Π').

Lemma 1: Assume that $\Pi \sim P$, $\Pi' \sim P$, and $\Pi \neq \Pi'$. Assume also that $A_1 \notin \Pi'$, A_1 is not totally dominant, and A_1 does not dominate any alphabet A_j, $j \neq 1$ that shares some A'_ℓ with A_1. Then, P_1 is memoryless.

Proof outline: Assume A_1 shares A'_ℓ with A_j. Let $a \in A_1 \cap A'_\ell$, and $s \in \mathcal{S}(P_1)$. The assumption that $A_1 \not\sqsupset A_j$, and the properties summarized in Fact 1 guarantee the existence of strings $U, V \in \Pi^*$ and $u, v \in \mathcal{A}^*$ such that u is consistent with U, $u[A_1] = \tilde{u}s$ for some $\tilde{u} \in A_1^*$, A_1 does not occur in V, $P_{\rm w}(A_1|UV) > 0$, v is consistent with V and is independent of s, and $|v[A_j \cap A'_\ell]| \geq \mathrm{ord}(P'_\ell)$. Computing probabilities according to the two given IMP representations, $\mathcal{I}_\Pi$ and $\mathcal{I}_{\Pi'}$, of P, we obtain

$$P(a|uv) = P_1(a|s)P_{\rm w}(A_1|UV) = P'_\ell(a|v[A'_\ell])P'_{\rm w}(A'_\ell|U'V'),$$

where $U'V' = \mathbf{A}_{\Pi'}(uv)$. It follows that $P_1(a|s)$ is independent of s. Using similar tools, and relying also on the fact that A_1 is not totally dominant, this independence can be proved also for the case where $a \in A'_\ell \in \Pi'$ and $A'_\ell \subseteq A_1$, thus establishing the fact that P_1 is memoryless. ∎

Corollary 1: Assume that $\Pi \sim P$, $\Pi' \sim P$, $\Pi \neq \Pi'$, $A_1 \notin \Pi'$, and A_1 is not dominant. Then, P_1 is memoryless.

In the sequel, we assume that $P_{\rm w}$ is such that no two alphabets in Π are in mutual domination. As discussed in Section III-A, this ensures that $\sqsupset$ defines a strict partial order on Π. We classify alphabets in Π into layers L_i, $i \geq 0$, using the following procedure, which starts with $i=0$ and $\hat{\Pi}=\Pi$:

1) Let L_i consist of all the alphabets in $\hat{\Pi}$ that do not dominate other alphabets in $\hat{\Pi}$ (i.e., the minimal elements in $\hat{\Pi}$ for the partial order $\sqsupset$). Since $\hat{\Pi}$ is finite and $\sqsupset$ is a strict partial order, L_i is not empty.
2) Let $\hat{\Pi} = \hat{\Pi} \setminus L_i$.
3) If $\hat{\Pi} \neq \phi$, increment i and go to Step 1. Otherwise, stop.

Theorem 1: Assume that, for an IMP $P = \mathcal{I}_\Pi$,

i) no two alphabets in Π are in mutual domination,
ii) no alphabet in Π is totally dominant, and
iii) none of the processes P_i is memoryless.

Then, if $P = \mathcal{I}_{\Pi'}$ for some partition Π', we must have $\Pi = \Pi'$.

Proof: Let r denote the largest index i attained in Step 3 above, i.e.,

$$\Pi = L_0 \cup L_1 \cup \cdots \cup L_r\,. \quad (2)$$

We prove, by induction on i, that $L_i \subseteq \Pi'$ for $0 \leq i \leq r$. By the definition of L_0, alphabets $A_i \in L_0$ are not dominant. Thus, by Corollary 1, we must have $A_i \in \Pi'$, since A_i is not memoryless by assumption (iii). Hence, $L_0 \subseteq \Pi'$.

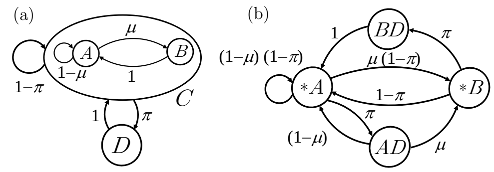

Fig. 1. Switches for non-unique IMP representation.

Assume now that the induction claim has been proven for $L_0, L_1, \ldots, L_i$, $i < r$. Let A_j be any alphabet in L_{i+1}. It follows from the definition of the layers $L_{i'}$ that A_j only dominates alphabets in layers $L_{i'}$, $i' \leq i$. By our induction hypothesis, alphabets in these layers are elements of Π', and, thus, they do not share with other alphabets. Thus, A_j does not dominate any alphabet A_h with which it shares any A'_ℓ. By Lemma 1, we must have $A_j \in \Pi'$, since A_j is neither totally dominant nor memoryless by the assumptions of the theorem. Hence, $L_{i+1} \subseteq \Pi'$, and our claim is proven. Thus, it follows from (2) that $\Pi \subseteq \Pi'$, and, since both Π and Π' are partitions of the same alphabet $\mathcal{A}$, we must have $\Pi = \Pi'$. ∎

Example 1: We consider alphabets A, B, D, and $C = A \cup B$, and respective associated processes P_A, P_B, P_D, P_C. Part (a) of Fig. 1 shows a switch $P_{\rm w}$ of order 1 over $\Pi = \{C, D\}$. Here, P_C is in itself an interleaved process $P_C = \mathcal{I}_{\{A,B\}}(P_A, P_B; P_{\rm w}^C)$ with $P_{\rm w}^C(A|A) = 1-\mu$, $P_{\rm w}^C(B|A) = \mu$, $P_{\rm w}^C(A|B) = 1$, for $\mu \in (0,1)$ and P_B chosen as a memoryless process so that P_C has finite memory. Part (b) shows a switch $P'_{\rm w}$ of order two over $\Pi' = \{A, B, D\}$. State $*A$ (resp. $*B$) represents all states that end in A (resp. B). It is readily verified that $P = \mathcal{I}_\Pi(P_C, P_D; P_{\rm w}) = \mathcal{I}_{\Pi'}(P_A, P_B, P_D; P'_{\rm w})$. C is totally dominant in $\mathcal{I}_\Pi$, and there are no memoryless processes nor mutually dominating alphabets in $\mathcal{I}_\Pi$; A, on the other hand, is totally dominant in $\mathcal{I}_{\Pi'}$.

C. Ambiguities due to memoryless components

In this subsection, we remove Condition (iii) of Theorem 1, while strengthening Conditions (i) and (ii) by excluding all forms of alphabet domination. We characterize all the representations of an IMP when ambiguities, if any, are due solely to memoryless components. The characterization generalizes that of [1] for a memoryless switch.

We say that a partition Π' is a *refinement* of Π if every subalphabet $A_i \in \Pi$ splits into one or more subalphabets in Π'. Let k be a nonnegative integer, and consider the sets $\mathcal{S} = \Pi^k$ and $\mathcal{S}' = (\Pi')^k$ (where, by convention, $\Pi^0 = (\Pi')^0 = \{\lambda\}$). When Π' is a refinement of Π, we define the mapping $\Psi_{\Pi,\Pi',k} : \mathcal{S}' \to \mathcal{S}$ as follows: for $k = 0$, we have $\Psi_{\Pi,\Pi',0}(\lambda) = \lambda$, and for $k > 0$ and all $B'_1 B'_2 \ldots B'_k \in \mathcal{S}'$, we have $\Psi_{\Pi,\Pi',k}(B'_1 B'_2 \ldots B'_k) = A_{i_1} A_{i_2} \ldots A_{i_k}$, where A_{i_j} is the unique element of Π such that $B_j \subseteq A_{i_j}$. We will omit some or all of the subscripts Π, Π', k of Ψ when clear from the context.

Lemma 2: Let $\Pi = \{A_1, A_2, \ldots, A_m\}$, and consider the refined partition $\Pi' = \{B_1, B_2, A_2, \ldots, A_m\}$ of $\mathcal{A}$ (i.e., $A_1 = B_1 \cup B_2$). Let $P = \mathcal{I}_\Pi(P_1, P_2, \ldots, P_m; P_{\rm w})$, where P_1 is memoryless, and let $P' = \mathcal{I}_{\Pi'}(P_1^{(1)}, P_1^{(2)}, P_2, \ldots, P_m; P'_{\rm w})$, where both $P_1^{(1)}$ and $P_1^{(2)}$ are memoryless. Then, there exist initial state assignments such that $P = P'$ if and only if the following conditions hold:

$$P_1^{(j)}(b) = \frac{P_1(b)}{P_1(B_j)}, \quad b \in B_j\,, \quad j \in \{1,2\}, \tag{3}$$

$$\mathcal{S}(P'_{\mathrm{w}}) = \{\, S' \in (\Pi')^{k_{\mathrm{w}}} \,\big|\, \Psi(S') {\in} \mathcal{S}(P_{\mathrm{w}}) \}, \tag{4}$$

where $k_{\mathrm{w}}{=}\mathrm{ord}(P_{\mathrm{w}})$, and for all $A{\in}\Pi'$ and $S'{\in}\mathcal{S}(P'_{\mathrm{w}})$, with $S{=}\Psi(S')$,

$$P'_{\mathrm{w}}(A|S') = \begin{cases} P_{\mathrm{w}}(A|S), & A = A_i,\ i \geq 2, \\ P_{\mathrm{w}}(A_1|S)P_1(B_j), & A = B_j,\ j = 1, 2\,. \end{cases} \tag{5}$$

Lemma 2 is interpreted as follows: since, given $\mathcal{I}_\Pi$, $P_1^{(1)}, P_1^{(2)}$, and P'_{w} can always be defined to satisfy (3)–(5), an IMP P with a nontrivial memoryless component always admits alternative representations where the alphabet associated with the memoryless process has been split into disjoint parts (the split may be into more than two parts, if the lemma is applied repeatedly). We refer to such representations as *memoryless refinements* of the original representation $\mathcal{I}_\Pi$. Using the lemma repeatedly, we conclude that P admits a refinement where all the memoryless components are defined over singleton alphabets. We will refer to this representation as *maximally refined*. On the other hand, the memoryless components $P_1^{(1)}$ and $P_1^{(2)}$ of P' can be merged if and only if P'_{w} satisfies the constraint

$$P'_{\mathrm{w}}(B_2|S') = \gamma P'_{\mathrm{w}}(B_1|S') \tag{6}$$

for a constant γ independent of $S' \in \mathcal{S}(P'_{\mathrm{w}})$. When (6) holds, we set $P_1(B_1) = 1/(1+\gamma)$ and $P_1(B_2) = \gamma/(1+\gamma)$, and P_1, P_{w} are defined implicitly by (3)–(5). Notice that the constraint (6) is trivially satisfied when the switch is memoryless. Thus, in this case, memoryless component processes can be split or merged arbitrarily to produce alternative IMP representations, as noted in [1]. When the switch has memory, splitting is always possible, but merging is conditioned on (6). We refer to a representation where no more mergers of memoryless processes are possible as *minimally refined*. We refer to the partition associated with a minimally refined representation as *canonical* (relative to P).

We say that the representations $\mathcal{I}_\Pi$ and $\mathcal{I}_{\Pi'}$ of an IMP P *coincide up to memoryless components* if the set of component processes of positive order is the same in both representations. The following lemma follows from the conditions imposed by Lemma 2 on representations that coincide up to memoryless components. The proof is given in the full paper.

Lemma 3: Let $\mathcal{I}_\Pi$ and $\mathcal{I}_{\Pi'}$ be IMP representations of a process P that coincide up to memoryless components, and such that both are minimally refined. Then, $\Pi = \Pi'$.

We denote the canonical partition associated with an IMP $P = \mathcal{I}_\Pi$ by $(\Pi)^*_P$. Also, we say P is *domination-free* if there is no alphabet domination in *any* IMP representation of P.

Theorem 2: Let $P = \mathcal{I}_\Pi$ and $P' = \mathcal{I}_{\Pi'}$ be IMPs over $\mathcal{A}$. Assume, furthermore, that P and P' are domination-free. Then, $P = P'$ if and only if $(\Pi)^*_P = (\Pi')^*_{P'}$.

Proof: Assume $P = P'$. Since there are no dominant alphabets in either representation, it follows from Corollary 1 that the representations must coincide up to memoryless components. It then follows from Lemma 3 that $(\Pi)^*_P = (\Pi')^*_P = (\Pi')^*_{P'}$. The "if" part follows directly from Lemma 2 and the definition of a minimally refined partition. ■

Theorem 2 states that, in the domination-free case, all the IMP representations of a process are those constructible by sequences of the splits and mergers allowed by Lemma 2. The theorem extends the results in [1], since it reduces to the characterization of ambiguities therein when the switches are memoryless.

IV. Deinterleaving scheme

Let $P = \mathcal{I}_\Pi$ be an IMP with associated order vector $\mathbf{k} = (k_1, k_2, \ldots, k_m, k_{\mathrm{w}})$. Define $\alpha_i = |A_i|$. Similarly to [1], it can be shown that given Π and $\mathbf{k}$, there exists a *finite-state machine* (FSM) $\mathcal{F}_{\mathbf{k}}(\Pi)$ (in general, with infinite memory), with at most $Q(\Pi, \mathbf{k}) = \sum_{i=1}^m (\alpha_i - 1)\alpha_i^{k_i} + (m-1)m^{k_{\mathrm{w}}}$ free statistical parameters, such that $\mathcal{F}_{\mathbf{k}}(\Pi)$ generates P.

Given a finite alphabet A, a sequence $u^t \in A^t$, and a nonnegative integer k, denote by $\hat{H}_k(u^t)$ the kth order (unnormalized) empirical entropy of u^t, $\hat{H}_k(u^t) = -\log P_k^{\mathrm{ML}}(u^t)$, where $P_k^{\mathrm{ML}}(u^t)$ is the ML (or empirical) probability of u^t under a kth order Markov model with a fixed initial state.

Let z^n be a sequence over $\mathcal{A}$. We define the *deinterleaved sequence vector* $\mathcal{D}_\Pi(z^n) = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m, \mathbf{Z}_{\mathrm{w}})$, where $\mathbf{z}_i = z^n[A_i]$, $\quad 1 \leq i \leq m$, and $\mathbf{Z}_{\mathrm{w}} = \mathbf{A}_\Pi(z^n) \in \Pi^n$. Given the partition Π, and an order vector $\mathbf{k}$, we define the cost of z^n relative to Π and $\mathbf{k}$ as

$$\begin{aligned} C_{\Pi,\mathbf{k}}(z^n) \quad &= \quad \sum_{i=1}^{m} \Big(\hat{H}_{k_i}(\mathbf{z}_i) + \beta(\alpha_i - 1)\alpha_i^{k_i} \log n \Big) \\ &\qquad + \hat{H}_{k_{\mathrm{w}}}(\mathbf{Z}_{\mathrm{w}}) + \beta(m-1)m^{k_{\mathrm{w}}} \log n\,, \end{aligned} \tag{7}$$

where β is a nonnegative (penalization) constant. For convenience, we set the penalty terms in (7) all proportional to $\log n$, rather than the term corresponding to $\mathbf{z}_i$ being proportional to $\log |\mathbf{z}_i|$. Given our assumptions on P_{w}, $|\mathbf{z}_i|$ will, almost surely, be proportional to n, so this simplifying assumption does not affect the main asymptotic results.

Given a sample z^n from an IMP P, our deinterleaving scheme estimates a partition $\hat{\Pi}(z^n)$, and an order vector $\hat{\mathbf{k}}(z^n)$, for the estimated IMP representation of P. The desired estimates are obtained by the following rule:

$$(\hat{\Pi}(z^n), \hat{\mathbf{k}}(z^n)) = \arg\min_{(\Pi', \mathbf{k}')} C_{\Pi', \mathbf{k}'}(z^n), \tag{8}$$

where $(\Pi', \mathbf{k}')$ ranges over all pairs of partitions of $\mathcal{A}$ and order vectors $\mathbf{k}'$. In the minimization, if $C_{\Pi',\mathbf{k}'}(z^n) = C_{\Pi'',\mathbf{k}''}(z^n)$, for different pairs $(\Pi', \mathbf{k}')$ and $(\Pi'', \mathbf{k}'')$, the tie is broken first in favor of the partition with the smallest number of alphabets. Notice that although the search space in (8) is defined as a Cartesian product, once a partition Π' is chosen, the optimal process orders k'_j are determined independently for each $j \in \{1, 2, \ldots, m, \mathrm{w}\}$, in a conventional penalized ML Markov order estimation procedure (see, e.g., [5]). Also, it is easy to verify that the optimal orders $\hat{k}_j$ must be $O(\log n)$, reducing the search space for $\mathbf{k}'$ in (8).

Theorem 3: Let $P = \mathcal{I}_\Pi$. Then, for suitable choices of the penalization constant β, $\hat{\Pi}(z^n)$ is compatible with P, and $\hat{\mathbf{k}}(z^n)$ reproduces the order vector of the corresponding IMP representation $\mathcal{I}_{\hat{\Pi}}$, almost surely as $n \to \infty$. Furthermore, if P is domination-free, we have

$$\hat{\Pi}(z^n) = (\Pi)^*_P \quad \text{a.s.} \quad \text{as } n \to \infty\,.$$

Theorem 3 states that our scheme, when presented with a sample from an interleaved process, will almost surely recover an alphabet partition compatible with the process. If the interleaved process is domination-free, the scheme will recover the canonical partition of the process, from which *all* compatible partitions can be generated via repeated applications of Lemma 2.

The following lemma will be useful in proving the first claim of Theorem 3. The proof, omitted here, follows along lines similar to those of the proof of Theorem 2 in [1].

Lemma 4: Let $P = \mathcal{I}_\Pi$, and let $\mathbf{k} = (k_1, k_2, \ldots, k_m, k_\mathrm{w})$ be the corresponding order vector. Let Π' be a partition of $\mathcal{A}$ incompatible with P, and $\mathbf{k}'$ an arbitrary order vector of dimension $|\Pi'| + 1$. Then, for a sample z^n from P, and for any choice of $\beta \geq 0$, we have

$$C_{\Pi',\mathbf{k}'}(z^n) > C_{\Pi,\mathbf{k}}(z^n) \quad \text{a.s.} \quad \text{as } n \to \infty\,.$$

The following lemma, in turn, will be useful in establishing the second claim of Theorem 3.

Lemma 5: Let $\Pi, \Pi', \mathcal{I}_\Pi$ and $\mathcal{I}_{\Pi'}$ be as defined in Lemma 2, so that $\mathcal{I}_{\Pi'}$ is a memoryless refinement of $\mathcal{I}_\Pi$. Let $\mathbf{k} = (0, k_2, \ldots, k_m, k_\mathrm{w})$ be the order vector corresponding to $\mathcal{I}_\Pi$, and $\mathbf{k}' = (0, 0, k_2, \ldots, k_m, k_\mathrm{w})$ that of $\mathcal{I}_{\Pi'}$. For a sample z^n from P, and an appropriate choice of β, we have: if $k_\mathrm{w} > 0$, then

$$C_{\Pi',\mathbf{k}'}(z^n) > C_{\Pi,\mathbf{k}}(z^n) \quad \text{a.s.} \quad \text{as } n \to \infty\,, \tag{9}$$

while if $k_\mathrm{w} = 0$, then

$$C_{\Pi',\mathbf{k}'}(z^n) = C_{\Pi,\mathbf{k}}(z^n)\,. \tag{10}$$

Proof outline: Since Π' is a refinement of Π, we have $P^{\mathrm{ML}}_{\Pi'}(z^n) \geq P^{\mathrm{ML}}_{\Pi}(z^n)$, where P^{ML}_{Π} and $P^{\mathrm{ML}}_{\Pi'}$ denote maximum likelihood probabilities with respect to the representations $\mathcal{I}_\Pi$ and $\mathcal{I}_{\Pi'}$, respectively. It is not hard to see that $P^{\mathrm{ML}}_{\Pi}(z^n)$ coincides with the ML probability of yet another representation of the same process, namely one with partition Π', same component processes as $\mathcal{I}_{\Pi'}$, but a switch process with state set $\mathcal{S}(P_\mathrm{w})$ (where states of $\mathcal{S}(P'_\mathrm{w})$ have merged according to the mapping Ψ) and such that the ratio of the conditional probabilities of B_1 and B_2 is independent of the conditioning state. This model has the same number of parameters as $\mathcal{I}_\Pi$. Thus, the comparison between $C_{\Pi',\mathbf{k}'}(z^n)$ and $C_{\Pi,\mathbf{k}}(z^n)$ is equivalent to a comparison between penalized ML probabilities for two switch models on alphabets of size $m+1$, one which is Markov of order k_w, and one with the above merger in the state set and the additional constraint on the conditional probabilities of B_1 and B_2, when the true process is on the merged state set and does satisfy the additional constraint. The Markov process of order k_w is therefore refined in two ways to create the alternative (true) model, and the penalized ML comparison corresponds to an MDL-like test of the two models. When $k_\mathrm{w} = 0$, the refinement is trivial, implying (10). When $k_\mathrm{w} > 0$, the first type of refinement (i.e., on the state set) is addressed in [6], where the strong consistency of this type of test is shown. The second type of refinement (i.e., ignoring a constraint on the conditional probabilities) can be analyzed with similar tools, yielding (9). ■

Proof outline for Theorem 3: The first claim of the theorem is proved along the same lines as the proof of Theorem 3 in [1], separating the error event into two categories: one involving a bounded number of erroneous hypotheses, and one in which the number of parameters in each erroneous hypothesis is large. The first category is handled by Lemma 4, whereas the second category is handled as in [6]. The second claim of the theorem is proved by applying Lemma 5, which implies that in the domination-free case, the canonical partition beats other compatible partitions with more sub-alphabets. When $k_\mathrm{w}{>}0$, this follows from (9), while when $k_\mathrm{w}{=}0$, it follows from (10) and our tie-breaking convention. ■

V. Experimental results

n	(a)	(b)
500	8.5%	48.0%
1000	54.0%	96.0%
2500	99.5%	100.0%
5000	100.0%	100.0%
10000	100.0%	100.0%

Fig. 2. Experimental results

We report on experiments showing the practical performance of the proposed deinterleaver. In the experiments, we measured deinterleaving success ratio for sequences of various lengths. For each length, 200 sequences were tested. Each sequence was generated by an IMP with $m{=}3$, sub-alphabet sizes $\alpha_1{=}4$, $\alpha_2{=}5$, $\alpha_3{=}6$, component Markov processes of order one with randomly chosen parameters, and a switch of order one with uniform single-symbol marginal distribution. We compare results for two variants of the proposed scheme. Variant (a) implements (8) via exhaustive search over all partitions. Since this is rather slow, variant (b) uses a randomized gradient descent-like heuristic, which is much faster, and achieves virtually the same performance (for shorter sequences, the heuristic sometimes finds the correct partition even when it is not the one that minimizes cost; this explains the slightly better performance compared to the exhaustive search). Figure 2 lists the percentage of sequences correctly deinterleaved by each variant for each sequence length.

Acknowledgment. W. Szpankowski's work was partially done while visiting HP Labs, Palo Alto, CA, and also supported by NSF STC Grant CCF-0939370.

References

[1] G. Seroussi, W. Szpankowski, and M. J. Weinberger, "Deinterleaving Markov processes via penalized ML," in *ISIT*, 2009, pp. 1739–1743.

[2] T. Batu, S. Guha, and S. Kannan, "Inferring mixtures of Markov chains," in *COLT*, 2004, pp. 186–199.

[3] N. Landwehr, "Modeling interleaved hidden processes," in *ICML '08: Proceedings of the 25th International Conference on Machine Learning*. New York, NY, USA: ACM, 2008, pp. 520–527.

[4] R. P. Stanley, *Enumerative Combinatorics*. Cambridge: Cambridge University Press, 1997, vol. 1.

[5] I. Csiszár and P. C. Shields, "The consistency of the BIC Markov order estimator," *Annals of Stat.*, vol. 28, pp. 1601–1619, 2000.

[6] M. J. Weinberger and M. Feder, "Predictive stochastic complexity and model estimation for finite-state processes," *Journal of Statistical Planning and Inference*, vol. 39, pp. 353–372, 1994.